

Doble Grado en Ingeniería Informática y Administración y
Dirección de Empresas
2019-2020

Trabajo Fin de Grado

“Mejora estética de imágenes mediante la
combinación de redes neuronales
convolucionales y algoritmos evolutivos.”

Juan Abascal Sánchez

Tutor/es

Miguel Ángel Patricio Guisado

Antonio Berlanga de Jesús

Colmenarejo, Julio 2019



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento
– No Comercial – Sin Obra Derivada**

RESUMEN

En los últimos tiempos se han estado utilizando algoritmos para mejorar la calidad estética de imágenes. Esta mejora ha sido llevada a cabo mediante la aplicación sistemática de filtros que transforman la imagen original en otra con mejores características. En el presente trabajo se presenta un nuevo enfoque para realizar esta tarea. Se utiliza una red neuronal convolucional (CNN) entrenada con el conjunto de fotografías AVA, que contiene alrededor de 255.00. A este clasificador se le añade un algoritmo evolutivo diferencial que permite encontrar los filtros óptimos para optimizar la puntuación estética de la imagen. La red neuronal será utilizada como función de ajuste del algoritmo evolutivo. Al final de la experimentación, se presenta la viabilidad de la metodología, analizando la convergencia y se añaden resultados visuales.

Palabras clave: Red Neuronal Convolucional, Algoritmo Evolutivo Diferencial, Mejora de Imágenes

ABSTRACT

Algorithms for image aesthetic improvement have been one of the applications more widely in the last years. The improvement of images has been made through the application of different filters that transform the original image into another whose aesthetics have been improved. In this work a new approach for the automatic improvement of the aesthetics of images is presented. This approach uses a Convolutional Neural Network (CNN) trained with the AVA photography dataset, which contains around 255,000 images that are valued by amateur photographers. Once trained, the algorithm will be able to assess an image in terms of its aesthetic characteristics. Through an evolutionary differential algorithm, an optimization process will be carried out in order to find the set of filters that improve the aesthetics of the original image. The trained CNN will be used as DE's fitness function. At the end of the experimentation, the viability of this methodology is presented, analyzing the convergence capacity and some visual results.

Keywords: Convolution Neural Network, Differential Evolution Algorithm, Aesthetic Image Improvement

TABLA DE CONTENIDO

1. Introduction	1
2. Estado del arte	5
2.1 Técnicas automáticas de mejora de imágenes	5
2.3 Datos y capacidad de procesamiento	7
2.4 Bases de datos de fotografía	8
3. Desarrollo y resultados	9
4.1 Componentes del programa	9
3.1.1 Preprocesamiento	10
3.1.2 Optimización.....	10
3.2 Etapa de optimización.....	11
3.3 Red Neuronal Convolucional.....	13
3.2.5 Limitaciones.....	20
3.3 Resultados	21
4. Conclusiones.....	28
5. Planning and Budgeting	30
Referencias	34

INDICE DE TABLAS

Tabla 1	19
Tabla 2	22
Tabla 3	24
Table 4	32
Table 5	32
Table 6	33

ÍNDICE DE FIGURAS

Figure 1. Most important features for smartphone's users [3]	2
Figura 2. Proceso que sigue el programa para realizar la mejora en la calidad estética de una imagen.....	10
Figura 3. Esquema del algoritmo diferencial evolutivo. [31].....	13
Figura 4. Explicación de los filtros convolucionales.....	14
Figura 5. Explicación de los filtros convolucionales.....	15
Figura 6 [20]. Arquitectura de la red utilizada en el trabajo.....	16
Figura 7. Ejemplo de distribución de puntuaciones para una imagen procesada por la red neuronal.	17
Figura 8. Evolución de los errores de entrenamiento y de validación.....	20
Figura 9. Distribución del tiempo de ejecución del algoritmo.	22
Figura 10. Relación entre mejora de la puntuación de las imágenes y puntuación original.....	23
Figura 11. Distribución de las puntuaciones antes (azul) y después (rojo) de ejecutar el algoritmo de optimización.	24
Figura 12. Diagrama de cajas que representa la distribución de los filtros aplicados para todas las imágenes.	25
Figura 13. Comparación de dos imágenes antes y después de aplicar el algoritmo.....	26
Figura 14. Comparación de dos imágenes antes y después de aplicar el algoritmo.....	26
Figura 15. Comparación de dos imágenes antes y después de aplicar el algoritmo.....	27
Figura 16. Comparación de dos imágenes antes y después de aplicar el algoritmo.....	27
Figure 17. Gantt diagram.....	31

1. INTRODUCTION

Photography is the art and technique of obtaining durable images using light-sensitive material. The first permanent photograph dates from 1826 and was obtained using a camera obscura and a mixture of Judea bitumen that was sensitive to light. Since then, the photography industry has expanded, not only in sales volume but in techniques used for taking images. Many of these techniques have been losing ground as other possibilities of taking photographs were discovered. In 1988 Fuji put on sale the first digital camera and these have been evolving to this day.

Nowadays, cameras, specially smartphone cameras, take advantage of artificial intelligence and automatic image enhancement methods to improve the quality of the pictures they take while having small and affordable lenses and sensors. The algorithm proposed in this project uses a combination of machine learning and an optimization algorithm, differential evolution to improve the aesthetic quality of images.

1.1 Socio-economic environment

Nowadays, digital camera sales corresponds to 80.3% of photographic products sales in China [1], while the film cameras only correspond to 1.6% of the sales of the industry. The revenues of the industry have been falling since 2012 [1]. Regarding the global market, the digital camera market reached a figure of \$ 18,600 million dollars in 2017. The compound annual growth rate of 11.5% is expected to decrease during the period between 2018 and 2023 [2]. This decrease in the digital camera market is significantly favored by the availability of new features in the "smartphones". The cameras of these devices are improving year after year and consumers are leaving aside traditional digital cameras [2]. In addition, consumers believe that a "smartphone" is a better investment because it offers more features.

Smartphone's camera quality has progressively improved until they are close to the quality of digital SLR cameras. Although in such small devices is not possible to replicate the entire lens system of a SLR camera, the advances in artificial intelligence and digital image enhancement techniques have allowed smartphones to be worthy competitors of the camera industry. Digital cameras have a greater potential especially for users who are familiar with the photography and processing techniques available today. Smartphones try to imitate the processing of professional photographers to allow users to obtain better

quality photographs without having the knowledge about the different post-processing techniques.

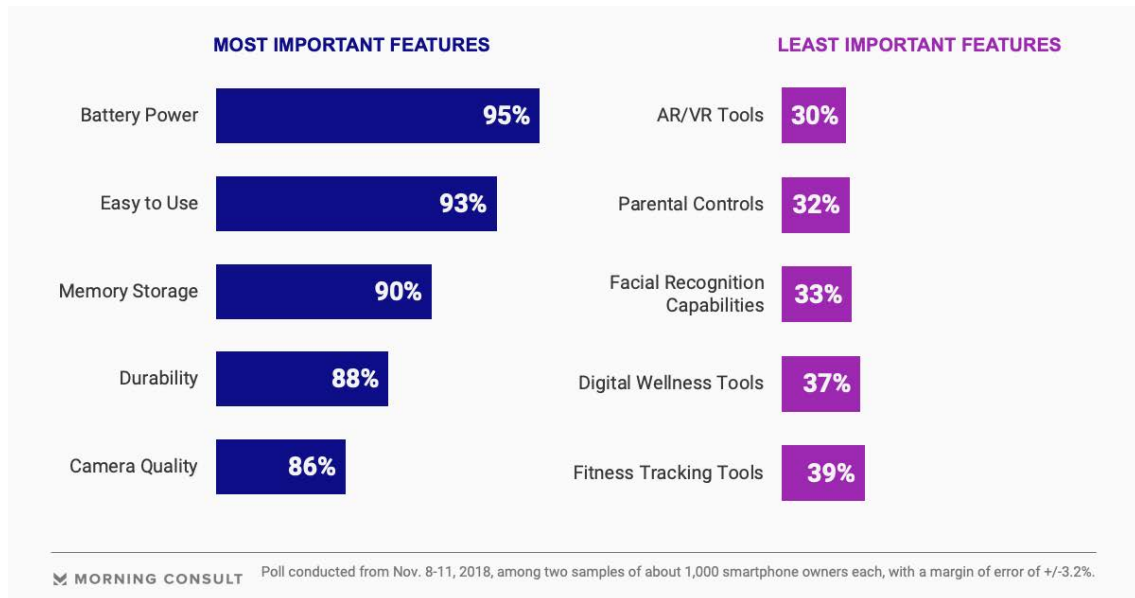


Figure 1. Most important features for smartphone's users [3]

The camera is one of the top 5 priorities for consumers when buying a smartphone [2]. That is why the manufacturers of "smartphones" devote many resources in the development of the cameras of their devices. Hence, smartphone's cameras have improved so much in recent years. Many of the advances have been implementation of the HDR or the portrait mode option, but in many cases the pictures post could be improved.

This work focuses on the proposal of a new method of image aesthetic improvement by choosing optimal post-processing parameters to. In addition, the viability of the method to be incorporated into state-of-the-art smartphones will be studied or if on the contrary this technique requires too much computing time. One of the great advantages that the proposed method provides is the ability to apply filters that adapt to the needs of the image.

The improvement of the quality of the images could benefit smartphones' manufacturers a competitive advantage in front of the rest of its competitors since it is one of the most important features for consumers.

1.2 Legal framework

It is important for any project that is going to process or store information to consider the regulatory framework that protects consumers of the final product that can use the proposed algorithm. Satisfy with the provisions of current legislation is essential, but it is also advisable to comply with good ethical and moral practices that govern the profession of engineering.

The current legislation that regulates the processing of personal data is made up of Regulation (EU) 2016/679 of the European Parliament and of the Council of April 27, 2016, regarding the protection of natural persons and the processing of their personal data. [4] and by Organic Law 3/2018, of December 5, on the Protection of Personal Data and guarantee of digital rights [5]. It must be taken into account that the image improvement algorithm does not need to gather information about the user because all the information processed by this algorithm might be in the smartphone's local environment.

If the algorithm is used to make improvements of images outside the device, it must comply with what is stipulated in the current legislation. The user must give explicit consent to the use and storage of their data. The use of the data must be lawful, loyal and transparent. Regulation (EU) 2016/679 stipulates penalties up to 20,000,000 euros or 4% of the global annual turnover to those companies that commit very serious infractions.

Although there are no current specific regulations for artificial intelligence and machine learning algorithms, the European Commission has issued a Communication carrying out a study on Artificial Intelligence in Europe. As established by the European Commission in its Communication COM (2018) 237 [6] on Artificial Intelligence for Europe, a legislation that regulates AI and ML will arrive soon. As soon as legislation is passed, the proposed algorithm must be reviewed and modified to comply with the provisions of the applicable regulations.

1.3 Objectives

The main objective of this work is the development of an algorithm that uses the combination of convolutional neural networks and differential evolution algorithms to improve the aesthetic quality of images. It will be desirable that the improvement of images is made by using simple photographic filters, so the algorithm is not dependent on these filters. In addition, the intensity of each filter will be different for each image, creating an algorithm that can make improvements regardless of the input image.

As a secondary objective it will be studied the implementation of the algorithm in smartphone devices. Given the current importance of smartphones in the world of digital photography, it is interesting to implement the optimization algorithm in a mobile device. The study will clarify whether the algorithm developed can be used by smartphones in real time.

1.4 Document structure

The document is divided into sections and subsections in which information is provided on the subject studied and the theme of this work is developed.

Firstly, in the objectives section, the goals of the study are presented. The objectives focus on the academic contribution of the work.

Secondly, a review of the previous literature and how other authors have faced similar problems is provided. It summarizes and specifies its solutions that help to understand and support to develop this work.

Thirdly, the technique that is going to be used to solve the problem is presented and the implementation of the solution is detailed. The results obtained are also presented and an analysis of these is made.

Finally, the conclusions that have been obtained from the experiments are made and further investigation is proposed for future work. It leaves a path open in the investigation of related topics.

2. ESTADO DEL ARTE

2.1 Técnicas automáticas de mejora de imágenes

Diferentes estudios han tratado de mejorar la iluminación de fotografías y la compensación de color [7], tratan de arreglar la diferencia que existe entre la visión humana y las imágenes tomadas por las cámaras de fotos. Mientras que en las fotografías existen zonas de baja luz con muy poca información, los seres humanos son capaces de ver toda la escena perfectamente expuesta. Por ello centran sus esfuerzos en realizar realces de luz en las zonas con baja luminosidad.

Otros estudios intentan mejorar el contraste de las imágenes. Una de las técnicas más famosas es la ecualización del histograma [8]. Esta técnica realiza una transformación sobre el histograma de una imagen para intentar conseguir un histograma uniforme que maximice el contraste sin perder información de tipo estructural, es decir manteniendo los bordes y las formas presentes en la imagen. Hay versiones más avanzadas de la ecualización del histograma [9] que emplea para el mapeo la información de un área alrededor de cada píxel. Esta técnica ayuda especialmente con imágenes en las que el contraste varía por zonas.

Para la mejora automática del color se utilizan técnicas diferentes. Los atributos del color son el tono, la saturación y la intensidad. Diversos estudios aconsejan no modificar el tono de las imágenes ya que podría distorsionar la imagen. Por ejemplo, los tonos naranjas de la piel podrían convertirse en tonos verdes. La saturación mide la pureza del color, por lo tanto, un color muy saturado es un color muy vivo. La intensidad mide la concentración de un color. Algunos trabajos como el de Kumar et al. [10] y Song et al. [11] utilizan espacios de color diferentes al RGB para realizar los cambios necesarios en el color de la imagen sin afectar al tono de los colores.

2.2 Técnicas de mejora de imágenes basadas en redes neuronales

El reciente auge de las redes de neuronas ha abierto un nuevo espacio de investigación para realizar mejoras estéticas de imágenes. A continuación, se muestra un pequeño repaso de esta técnica.

Desde el año 2012, cuando fue creada AlexNet [12] para competir en el *ImageNet Large Scale Visual Recognition Challenge*, las redes neuronales convolucionales profundas han cambiado el panorama de la inteligencia artificial. El área de aplicaciones de este tipo de

redes es muy amplia, y se utilizan para reconocer textos escritos a mano, análisis semántico de fotografías y reconocimiento de personas. En algunos casos han superado el rendimiento a la hora de realizar diferentes tareas.

Uno de los mayores inconvenientes de implementar soluciones basadas en redes neuronales convolucionales es que son muy intensivas en computación y en datos [13], por lo que se requiere mucho tiempo de entrenamiento para alcanzar la precisión deseada. El entrenamiento de una red neuronal profunda como GoogleLeNet que tiene 6.797.700 parámetros y 1.502.000.000 operaciones se puede realizar en una semana utilizando varias GPUs de forma paralela [14].

También se han realizado otros trabajos que han llevado las redes de neuronas a otros campos como el arte. Hoy en día, se pueden encontrar redes que han sido entrenadas para dibujar cuadros con el estilo de Vicent van Gogh [15] o componer canciones que se asemejan a las melodías compuestas por el autor alemán Johann Sebastian Bach [16]. Sin embargo, en otros campos menos definidos como la fotografía el aprendizaje automático no ha tenido tanto éxito.

Diferentes trabajos [17-19] han tratado de resolver el problema de evaluación de calidad de imágenes. Estos trabajos han proporcionado distintas alternativas para la resolución de dicha cuestión. Muchos de estos trabajos han tratado de solucionar el problema realizando una clasificación binaria para determinar si una imagen es de buena o de mala calidad.

También existen trabajos que ofrecen otro enfoque como NIMA [20], en el que se consigue una medida de calidad de la imagen simulando las puntuaciones en el rango del uno al diez que una imagen recibiría en un concurso de fotografía. En este caso, el output de la red neuronal es una distribución de la cual se puede extraer la media y la desviación estándar para ofrecer una valoración cuantitativa de la calidad estética de la imagen. El trabajo de Kao también basa su enfoque en conseguir una medida de calidad continua en lugar de binaria.

Otros trabajos se centran en la mejora estética de imágenes mediante la utilización de redes neuronales. El trabajo de Deng, Loy y Tang [21] utiliza “adversarial learning” para mejorar la calidad estética de fotografías. Utilizan un algoritmo poco supervisado y consiguen mejorar el color de las imágenes. Sus resultados de edición son similares a los realizados por seres humanos.

Hui y Fang [22] introducen el concepto de filtros ortogonales para poder mejorar filtros específicos de las imágenes. Por ejemplo, introducen un filtro de saturación que únicamente entiende de saturación. Crean por tanto una red neuronal que se encarga de evaluar cada filtro. Además, utilizan redes de neuronas adversarias para generar filtros que no existen todavía. Son capaces de generar una máscara de luminosidad que se aplica de manera diferente a las distintas partes de la imagen, imitando el retoque que haría un fotógrafo profesional.

2.3 Datos y capacidad de procesamiento

El entrenamiento de redes neuronales profundas es muy intensivo en datos y en poder computacional [13]. La aparición de tendencias como el Big Data ha ayudado a la recolección de un gran número de datos que tienen una alta disponibilidad, mucho mayor que la que tendrían con software y herramientas tradicionales. El número de datos que son procesados en Internet ha aumentado enormemente en los últimos 10 años y se espera que siga creciendo [23].

Otra tendencia que está influyendo en la recolección de datos a nivel mundial es el *IoT* (*Internet of Things*). El número de dispositivos interconectados no ha parado de crecer desde el año 2015, pasando de 3.8 billones de dispositivos a 8.3 billones en 2019 [24]. La cantidad de datos generada por estos dispositivos permiten entrenar nuevos modelos de aprendizaje automático.

La capacidad de procesamiento de las *GPUs* (*graphics processing unit*) ha hecho posible reducir el tiempo requerido para entrenar redes de neuronas convolucionales. Las GPUs están optimizadas para realizar cálculos matriciales que son necesarios en cualquier red neuronal [13]. La optimización de este tipo de hardware en los últimos tiempos y especialmente su alta disponibilidad a través de servicios cloud ha contribuido a la democratización del aprendizaje automático con redes neuronales profundas.

Actualmente están apareciendo nuevas unidades de procesamiento, *TPUs* (*Tensor Processor Units*), específicamente diseñadas para optimizar soluciones de aprendizaje automático. Este tipo de *hardware* desarrollado por la compañía americana Alphabet Inc, son entre 15 y 30 veces más rápidas que las GPUs tradicionales a la hora de inferir, mientras que su coste es mucho menor. Además, son muchos más energéticamente eficientes que las GPUs [25].

2.4 Bases de datos de fotografía

En la actualidad se pueden encontrar diferentes bases

- A Large-Scale Database for Aesthetic Visual Analysis (AVA) [26]: Contiene alrededor de 255.000 imágenes y están valoradas por fotógrafos aficionados. Cada imagen tiene asociada la frecuencia con la que los usuarios han seleccionado cada una de las puntuaciones del 1 al 10. Además, presentan metadatos relacionados con el tipo de concurso para el que la imagen fue tomada.
- Tampere Image Database 2013 [27]: Este dataset está formado por 3000 imágenes generadas a partir de 25 imágenes a las que se han aplicado diferentes tipos de filtros. Después han sido mostradas por pares y los encuestados han tenido que votar cuál de las dos imágenes les gustaba más. Así se ha conseguido dar una puntuación a cada imagen de 0 a 9.
- Aesthetics with Attributes Database (AADB) [28]: Contiene 10.000 imágenes que están distribuidas entre imágenes sacadas por profesionales e imágenes sacadas por fotógrafos aficionados. Además de contener la distribución de puntuaciones, AADB tiene etiquetas relacionadas con el valor estético de las fotografías como son: contenido interesante, color vivo o simetría.

3. DESARROLLO Y RESULTADOS

Es importante comenzar esta sección definiendo la métrica de valoración estética de imágenes que permite comparar dos o más fotografías. Esta métrica presenta un nivel de medida ordinal, ya que permite ordenar las fotografías de mayor a menor calidad estética o viceversa. Este hecho es una ventaja frente a un nivel de medida categórico con el que no es posible establecer un orden, ni calcular la mejora que sufre una imagen al ser editada.

La métrica de valoración estética o puntuación de una imagen es calculada mediante la red neuronal convolucional, cuyo output es un vector de diez dimensiones. Los valores de ese array representan el porcentaje de personas que votarían esa imagen con la puntuación igual a la posición del vector. La ecuación (1), que es una media ponderada, muestra cómo se calcula la puntuación a partir del vector.

$$s = \sum_{n=1}^N p_{sn} \times n \quad (1)$$

Además de la media, se puede calcular la desviación estándar de las puntuaciones para conocer la dispersión de las votaciones. Una puntuación una desviación estándar elevada representa que habría un menor consenso a la hora de dar las puntuaciones.

La red neuronal convolucional se encarga de calcular las puntuaciones de las imágenes, mientras que el algoritmo evolutivo permite realizar las ediciones que optimizan dicha puntuación. Este proceso puede ser realizado gracias a que la métrica de valoración estética de las imágenes es un nivel de medida ordinal.

4.1 Componentes del programa

El programa sigue un flujo lineal, en el que el objetivo es mejorar la calidad estética de las imágenes mediante edición de post procesamiento. Para conseguirlo, el algoritmo empleado permite encontrar la combinación óptima de filtros fotográficos a aplicar sobre una imagen dada. Este algoritmo de optimización tomará como función de evaluación el resultado obtenido de una red neuronal convolucional entrenada para determinar la calidad estética de una imagen. Esto son solo las dos partes más importantes del flujo de información del programa.

En este apartado se explican detalladamente el flujo del programa y las decisiones que se han tomado a la hora de desarrollarlos. En la figura se muestra la secuencia del proceso

del programa. Tanto el preprocesamiento como el post procesamiento se ejecutan una única vez, en cambio el algoritmo diferencial evolutivo se ejecuta tantas veces como sea necesario.

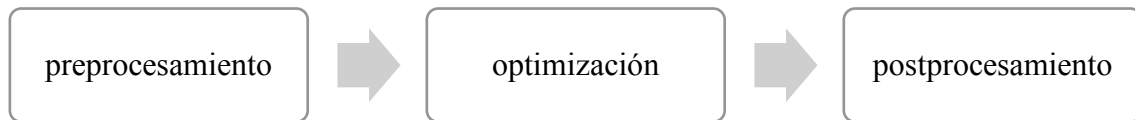


Figura 2. Proceso que sigue el programa para realizar la mejora en la calidad estética de una imagen.

3.1.1 Preprocesamiento

En la etapa de preprocesamiento se prepara la imagen para que pueda ser utilizada por el algoritmo evolutivo. Debido a las características de la red neuronal la imagen debe tener unas dimensiones de 224x224. Aunque la imagen sufra pérdida de información al convertirla de dimensiones, esto no es un impedimento para calcular la puntuación estética de la fotografía. Las modificaciones se aplican sobre la imagen original.

Existen diferentes maneras de realizar este preprocesamiento, recorte o redimensionamiento. Ambas técnicas presentan ventajas e inconvenientes. Por un lado, el recorte elimina información de la fotografía que puede afectar a la composición de esta, la información desechada podría ser crucial a la hora de determinar la puntuación de la fotografía. Por otro lado, el redimensionamiento también afecta a la composición de la imagen, pero esta vez en lugar de perderse información, la imagen se deforma alterando su aspecto ratio. Para este trabajo, se ha elegido la opción de redimensionamiento ya se prefiere que el algoritmo analice toda la información de la imagen, aunque esta pierda su aspecto ratio.

3.1.2 Optimización

La etapa del algoritmo evolutivo diferencial es la más compleja, y a su vez está compuesta por varias etapas. Esta etapa está explicada detalladamente en el apartado 3.2.

3.1.3 Post procesamiento

Una vez decidido cuál es la mejor combinación de filtros que se puede aplicar a la imagen, esta se procesa. La imagen procesada se almacena en la ruta especificada por el usuario.

3.2 Etapa de optimización

La etapa de optimización utiliza un algoritmo evolutivo diferencial para obtener la combinación de filtros que maximizan la puntuación estética de una imagen. La función objetivo del algoritmo evolutivo es una red neuronal convolucional.

3.2.1 Algoritmo evolutivo diferencial

Existen problemas en los que hay que optimizar las propiedades de un sistema mediante el ajuste de diferentes parámetros de este. Para comenzar a resolver este tipo de problemas computacionalmente, es necesario definir una función objetivo. En el caso del presente trabajo, la función objetivo no será definida formalmente si no que será el resultado del procesamiento de una red neuronal. El problema de optimización planteado se trata de un problema de minimización, ya que se intenta encontrar los valores del conjunto de parámetros que minimizan el resultado de la red neuronal.

El algoritmo elegido para llevar a cabo la tarea de optimización es un algoritmo evolutivo diferencial [29]. Se trata de una búsqueda directa que no necesita información acerca del gradiente de la función objetivo. El tipo estrategia elegida es usar el mejor vector como base, solamente un vector se utiliza para generar la población mutante y se utiliza una distribución uniforme “crossover” en la transformación de la población candidata.

Además, emplea un “criterio codicioso”, es decir que un nuevo conjunto de parámetros es aceptado cuando este reduce el valor de la función objetivo. Los algoritmos de optimización que utilizan este tipo de criterio tienden a encontrar mínimos locales en lugar de mínimos globales, por eso el método de optimización evolutivo diferencial ejecuta varios conjuntos de parámetros de manera simultánea.

El algoritmo evolutivo diferencial de este trabajo trata de optimizar cuatro filtros fotográficos que mejoran la calidad estética de las imágenes. Estas modificaciones se realizan sobre:

- Contraste: diferencia relativa entre los píxeles claros y oscuros de una imagen. Las imágenes con poco contraste son imágenes muy planas.
- Brillo: luminosidad de los tonos medios de una fotografía.
- Saturación: es la viveza del color. Una fotografía que tiene mucha saturación presenta colores vivos e intensos.
- Nitidez: nivel de acutancia de una imagen. Está relacionada con el contraste entre los bordes de los detalles y su fondo.

La librería empleada para la modificación de las imágenes permite dar valores entre 0 e infinito. Cuando el valor es uno la imagen se queda como está, cuando es 0, la imagen pierde todo el contraste, brillo saturación o nitidez. Se ha establecido que los valores que van a utilizarse a la hora de modificar las imágenes van de 0,8 a 3. Por lo tanto, el vector utilizado por el algoritmo evolutivo diferencial tiene 4 posiciones y los números entre los que oscila van de 0,8 a 3.

El algoritmo diferencial evolutivo es un proceso estocástico evolutivo que presenta cuatro etapas diferenciadas [30].

- **Inicialización:** Se crea la población de individuos. Cada individuo es una instanciación aleatoria del conjunto de parámetros. Además, se evalúa cada individuo con la función objetivo para ahorrar costes computacionales en el futuro. Este paso corresponde con los pasos 1 y 2 de la Figura 3.
- **Mutación y recombinación:** Para cada individuo de la población se seleccionan otros tres individuos diferentes de forma aleatoria, que son combinados para generar un individuo mutante. El individuo original y el individuo mutante son recombinados para crear el individuo candidato que será evaluado para ver su rendimiento en comparación con el individuo original. Este paso corresponde con los pasos 3 y 4 de la Figura 3
- **Evaluación y Reemplazo:** Se compara el resultado obtenido por el individuo candidato en la función objetivo con el resultado obtenido por el individuo original. Si el resultado del individuo candidato es mejor que el del original, este será sustituido en la población. Estos pasos corresponden con el paso 5 de la Figura 3.

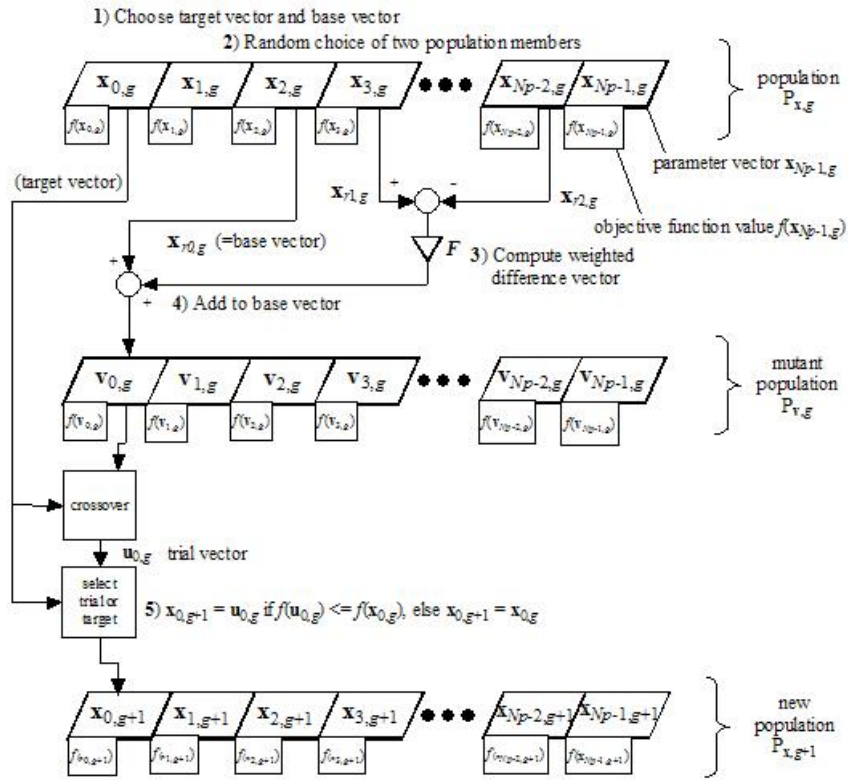


Figura 3. Esquema del algoritmo diferencial evolutivo. [31]

La función objetivo del algoritmo diferencial evolutivo viene dada por la puntuación que recibe una fotografía al aplicarse los parámetros de un individuo de la población. La puntuación de la imagen es un número entre 0 y 1, y el objetivo es maximizar ese valor. Como el algoritmo diferencial evolutivo realiza una tarea de minimización, hay que transformar el resultado producido por la red neuronal.

$$funcionObj = \min(1 - puntuación) \quad (2)$$

Por lo tanto, el algoritmo diferencial evolutivo consigue obtener el conjunto de parámetros que optimiza la puntuación de la imagen. La Figura 3 muestra este proceso.

3.3 Red Neuronal Convolutacional

La red neuronal convolutacional utilizada para resolver este problema está basada en el artículo Neural Image Assesment [20]. Para el entrenamiento de esta red se utiliza el conjunto de datos de fotografía AVA [26] que ha sido explicado en la sección 2. Además, la arquitectura empleada ha sido MobileNet [32], una arquitectura ligera diseñada para poder ser ejecutada en dispositivos móviles.

La elección de MobileNet frente a otras arquitecturas se debe a la posible inclusión de este algoritmo para utilizarse en la mejora de imágenes sacadas por dispositivos móviles. Esta arquitectura, aunque tiene una menor precisión que otras alternativas como ResNet [33] o VGGNet [34] presenta tiempos de inferencia mucho menor. El número de parámetros es mucho menor y por tanto el número de operaciones que se necesitan realizar es más bajo. Ante la situación de compromiso de elegir entre una mayor precisión o velocidad de inferencia, se ha elegido la segunda opción.

3.3.2 Arquitectura

La red de neuronas convolucional MobileNet, está formada por un tipo de capas convolucionales, “*Depthwise Separable Convolution*” que permite una mejora sustancial tanto en el número de operaciones como en el número de parámetros. A continuación, se compara el rendimiento de este novedoso tipo de convoluciones con las convoluciones tradicionales.

Una capa convolucional tradicional está formada por n filtros, estos filtros se deslizan a lo largo y ancho de las imágenes realizando productos vectoriales. El resultado de aplicar n filtros a una imagen de dimensiones $(w_1 \times h_1 \times 3)$ es una matriz en de dimensiones $(w_2 \times h_2 \times n)$. El proceso de generar el filtro se muestra en la Figura 4. El ancho y alto del resultado de la convolución depende de factores como el paso y el relleno que se le aplique al filtro. Si el paso es de 1 y el relleno es de 1, entonces $w_1 = w_2$ y $h_1 = h_2$. Este proceso se ilustra en la Figura 5.

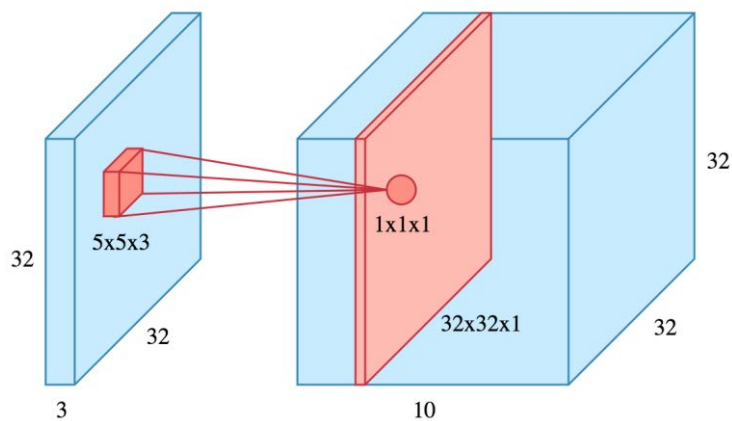


Figura 4. Explicación de los filtros convolucionales.

En la parte izquierda se observa la imagen de $(32 \times 32 \times 3)$ a la que se le aplica un filtro de tamaño $(5 \times 5 \times 3)$. El producto escalar genera un filtro de $(32 \times 32 \times 1)$ y en total la capa convolucional tiene 10 filtros por lo que el tamaño del resultado es de $(32 \times 32 \times 10)$. [35]

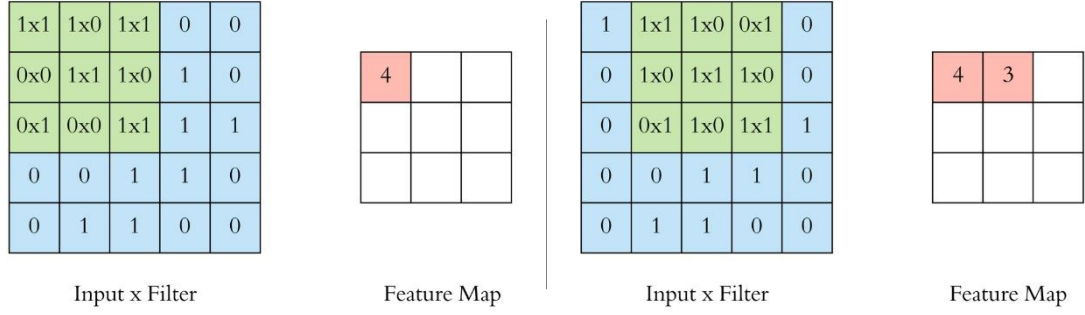


Figura 5. Explicación de los filtros convolucionales.
 Se observa el proceso que va realizando el filtro, en color verde y de dimensiones (3×3) para generar el mapa de características. En este caso el paso es de 1 pero el relleno es 0, por lo que $w_1! = w_2$ y $h_1! = h_2$. [35]

El número de operaciones necesarias para procesar una capa convolucional viene representado en la ecuación (3), donde D_k^2 es el tamaño del filtro, D_g^2 es el tamaño del mapa de características, M es el número de canales en la matriz de entrada y N es el número de filtros aplicados.

$$D_k^2 \times D_g^2 \times M \times N \quad (3)$$

MobileNet utiliza “*Depthwise Separable Convolution*”. Para utilizar este tipo de filtros, el proceso se divide en dos pasos. Primero se aplican “*depthwise convolutions*” que consiste en aplicar un filtro por cada canal de la matriz de entrada, y después se aplican “*pointwise convolutions*” que son N convoluciones de 1×1 para crear combinaciones lineales de la matriz resultado.

$$D_k^2 \times D_g^2 \times M \quad (4)$$

$$D_g^2 \times M \times N \quad (5)$$

La complejidad la parte de “*depthwise convolutions*” viene dada por la ecuación (4) mientras que la complejidad de “*pointwise convolutions*” viene dada por la ecuación (5). La complejidad total de una capa “*Depthwise Separable Convolution*” está representada en la ecuación (6).

$$D_k^2 \times D_g^2 \times M + D_g^2 \times M \times N \quad (6)$$

Si se compara el rendimiento de las capas convolucionales tradicionales y las “*Depthwise Separable Convolution*”, se obtiene la ecuación (7). La ganancia de rendimiento de las capas convolucionales utilizadas por MobileNet frente a las capas convolucionales tradicionales, necesitan entre 8 y 9 veces menos operaciones computacionales [32].

$$\frac{D_k^2 \times D_g^2 \times M}{D_k^2 \times D_g^2 \times M + D_g^2 \times M \times N} = \frac{1}{N} + \frac{1}{D_k^2} \quad (7)$$

Todas las capas de MobileNet, a excepción de la última capa, están seguidas de normalización en lote [36], que mejoran la velocidad de entrenamiento de las redes convolucionales profundas, y funciones de activación no lineales ReLU [12], que ayudan a la propagación hacia atrás de los gradientes en redes con muchas capas. Todo esto hace de la arquitectura MobileNet la red neuronal convolucional más eficiente en términos de complejidad computacional, y es idónea para sistemas con poder de procesamiento limitado.

A la arquitectura MobileNet, entrenada para el concurso ImageNet [12], se le han quitado las últimas dos capas de la red, que son capas no convolucionales. Estas dos últimas capas son un clasificador, que recibe las características extraídas por la parte convolucional de la red neuronal y da como resultado la probabilidad de pertenencia a una de las clases. Este comportamiento necesita ser modificado para adaptarse al problema del presente trabajo. Para ello se conectan dos nuevas capas que forman un nuevo clasificador, que recibe las características extraídas por la parte convolucional y genera una distribución de puntuaciones.

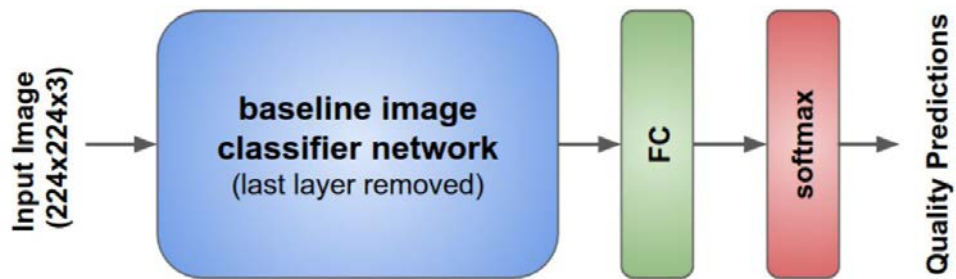


Figura 6 [20]. Arquitectura de la red utilizada en el trabajo.

La predicción de la red es una distribución de puntuaciones $p = [p_{s1}, p_{s2}, \dots, p_{sn}]$, donde sn es el porcentaje de votos que una imagen recibe para una puntuación dada. Cabe destacar que la suma de p es 1. Para ilustrar este concepto, se pone un ejemplo hipotético.

En la Figura 7 se observa una distribución en la que el 25% de los individuos que han votado han dado una puntuación de 6 puntos. Para calcular la puntuación total se emplea la ecuación (1). Como las puntuaciones en el conjunto de datos AVA van de 1 a 10, $n = 10$. La puntuación hipotética de la distribución de la imagen es de 5,42.

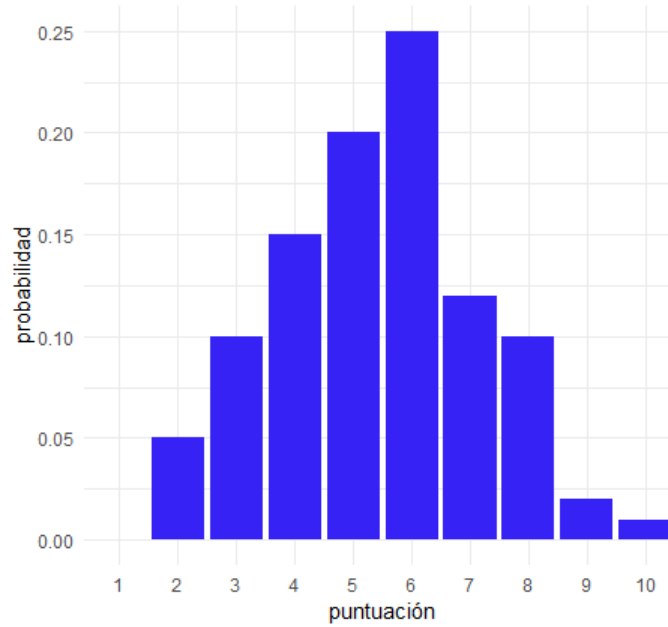


Figura 7. Ejemplo de distribución de puntuaciones para una imagen procesada por la red neuronal.

3.3.3 Función del error

La función del error en este trabajo mide la distancia que existe entre dos distribuciones univariadas. Por un lado, está la distribución que contiene el conjunto de datos que es la “distribución verdadera” y por otro lado está la distribución predicha por la red neuronal. Es necesario determinar una medida para poder comparar cuanto se ha equivocado la predicción de la red neuronal y ajustar los pesos de esta mediante el algoritmo de propagación hacia atrás [37]. Medidas típicas de los problemas de aprendizaje automático como el error cuadrático medio o entropía cruzada no pueden ser aplicadas para este problema.

Para comparar dos distribuciones se utiliza la medida de error *Earth Mover's Distance* (EMD) [38], que mide el coste mínimo de transformar una distribución en la otra. Esta

medida, por tanto, mide la exactitud del resultado de la red neuronal (\hat{p}) con la etiqueta de la imagen (p). El resultado proporcionado por la red neuronal

$$EMD(p, \hat{p}) = \left(\frac{1}{N} \sum_{k=1}^N |CDF_p(k) - CDF_{\hat{p}}(k)|^r \right)^{1/r} \quad (8)$$

Dónde $CDF_p(k)$ es la función acumulada de probabilidad para una imagen k . Esta medida también es conocida como métrica Wasserstein.

3.3.4 Entrenamiento

Debido a la falta de recursos y a la dificultad de entrenar una red neuronal profunda con tantos parámetros, se ha recurrido a la utilización de un modelo previamente entrenado por Christopher Lennan et al. [39] para resolver el problema de clasificación estética de imágenes.

El conjunto de datos AVA contiene 255.522 imágenes, de las cuales 229.971 han sido utilizadas como conjunto de entrenamiento y el resto, 25.551, han sido utilizadas como conjunto de validación.

Los pesos del modelo no han sido inicializados de forma aleatoria. Como punto de partida se ha utilizado MobileNet previamente entrenada para resolver ImageNet, ya que se considera que esa distribución de los pesos se parece más a la distribución buscada [20]. Esta técnica se denomina afinamiento de los pesos y consigue que sea posible reducir el tiempo de entrenamiento. Los hiperparámetros de entrenamiento están explicados en la Tabla 1. El entrenamiento se ha llevado a cabo en dos etapas en las que se han utilizado diferentes hiperparámetros.

En primer lugar, para evitar estropear los pesos ya entrenados sobre ImageNet, se ha entrenado únicamente la capa densa del modelo, es decir la parte que se ha añadido a la red entrenada, mientras que las capas convolucionales se mantienen constantes. Como los pesos de esta capa han sido inicializados aleatoriamente producirán un error alto, por lo que los pesos de la parte convolucional sufrirían una modificación grande. Como el objetivo de esta primera fase es aproximar los pesos de la última capa a la posible solución, se utiliza una tasa de aprendizaje elevada que permite hacer una búsqueda más amplia en el espacio de hipótesis. El impacto de la tasa de aprendizaje en la variación de

los pesos viene dado por la ecuación (9). La tasa de aprendizaje η controla cuanto se ajustan los pesos de la red respecto al gradiente del error.

$$w_{jk}^{(\tau+1)} = w_{jk}^{\tau} - \eta \times \frac{\partial E}{\partial w_{jk}^{\tau}} \quad (9)$$

La segunda etapa del proceso consiste en reentrenar todos los pesos de la red. En esta segunda etapa se busca hacer cambios pequeños en los pesos para que vayan convergiendo hacia un mínimo global. Además, se utiliza un decay para la tasa de aprendizaje que va haciendo que a medida que se entrena la red la tasa de aprendizaje va reduciéndose. Esta segunda parte utiliza los pesos entrenados previamente de manera más imprecisa y mediante pequeñas modificaciones consigue que el error se vaya reduciendo.

TABLA 1
PARÁMETROS UTILIZADOS PARA EL ENTRENAMIENTO DE LA RED NEURONAL.

Parámetro	Valor
Tamaño del batch	96
Tasa de aprendizaje I	0,001
Tasa de aprendizaje II	0,00003
Decay para tasa de aprendizaje II	0,000023
Tasa de dropout	0,75

En total para entrenar el modelo se han realizado 5 “epochs” completos con la primera de las configuraciones. Y otros 9 “epochs” con la segunda configuración. En total se han procesado 37.264 lotes de acuerdo con la ecuación (10). Al final del entrenamiento se ha conseguido un error de menos de 0,07 puntos, en ambos el conjunto de entrenamiento y el conjunto de prueba.

$$totalLotes = \frac{255.522}{tamañoLote} \times epochs \quad (10)$$

La evolución del error con respecto a los “epochs” se puede ver en la Figura 8. Se observa una mejora importante del primer al segundo “epoch” con una reducción de casi 0,02 puntos en el error. A partir del segundo “epoch” hasta el quinto la reducción del error es

mucho menor, casi insignificante esto se debe a que la función objetivo no se puede aproximar únicamente modificando los pesos del clasificador de la red neuronal. Es por ello por lo que para el sexto “epoch” se cambia el método de entrenamiento permitiendo que todos los pesos de la red neuronal se actualicen. Esto tiene un impacto notable en los “epochs” seis y siete, reduciéndose el error en otros 0,02 puntos. A partir del octavo “epoch” la disminución del error va disminuyendo hasta el “epoch” catorce, aunque a un ritmo menor. La gráfica del error indica que este podría seguir disminuyendo en el futuro. Aunque después de realizar los primeros “epochs”, había indicios de sobreentrenamiento, esta situación se corrige a partir del “epoch” ocho donde el error de validación es ligeramente superior al error de entrenamiento.

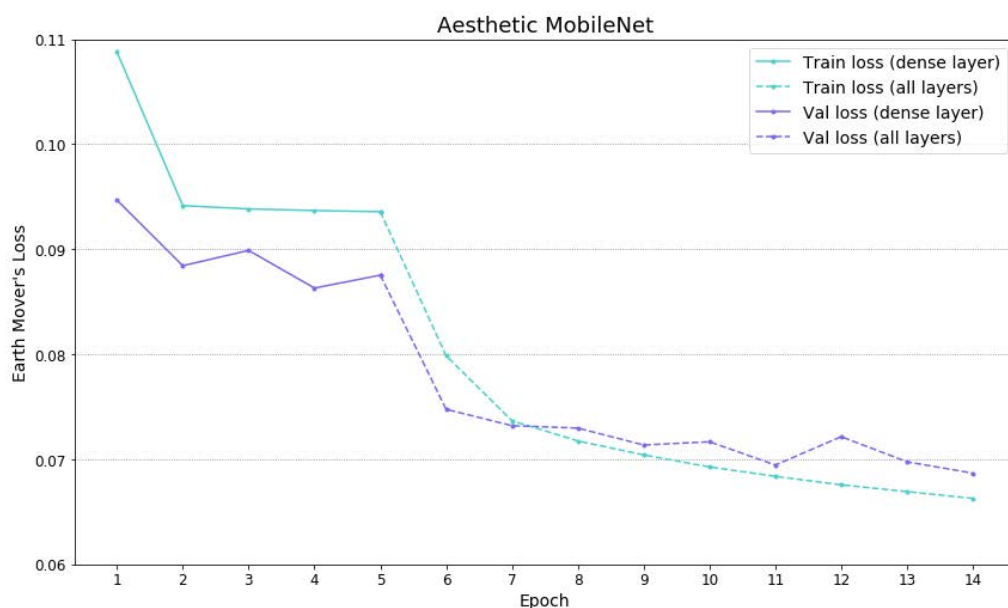


Figura 8. Evolución de los errores de entrenamiento y de validación.

El error de 0,068 significa que sería necesario mover un 0,068 de una distribución a la otra para conseguir dos distribuciones idénticas.

3.2.5 Limitaciones

Debido a que el conjunto de datos AVA presenta muchas imágenes con puntuación entre 4 y 6, la red neuronal tiene dificultades para predecir valores por encima de 7 y por debajo de 3. Esto genera un problema al diferenciar imágenes que sobresalen de la media y por tanto el algoritmo no es capaz de mejorarlas adecuadamente. Este comportamiento se puede ver en el gráfico de dispersión de la Figura 10.

Para solventar este problema sería necesario generar un conjunto de datos más equilibrado o realizar técnicas que permitan equilibrar AVA, como se explica en la sección 4.

3.3 Resultados

Para evaluar el algoritmo de optimización se ha utilizado un subconjunto de imágenes del conjunto de datos AVA. Se han seleccionado 1.000 imágenes de manera aleatoria que han sido procesadas por el algoritmo, almacenando tanto la puntuación original como la puntuación después de la optimización.

Es importante centrarse en varios aspectos diferentes para ser analizados:

- El tiempo de ejecución
- La mejora en la estética de las imágenes
- Los parámetros que más se modifican

En este estudio es importante conocer el tiempo que tarda el conjunto del algoritmo en ejecutarse. Si se desea poder implementar esta solución en cámaras de dispositivos móviles, el tiempo de ejecución tiene que ser rápido para no estropear la experiencia de usuario. El algoritmo utilizado en este proyecto requiere de mucho poder computacional. Se tiene que procesar la red neuronal muchas veces para que el algoritmo evolutivo diferencial pueda encontrar los mejores filtros.

Para conocer el tiempo que el algoritmo tarda en encontrar los filtros óptimos se ha sumado el tiempo que se tarda en generar a cada generación. La Tabla 2 muestra el tiempo medio que tarda el algoritmo en procesar las imágenes. Los valores están concentrados en torno a la media y el 50% de los experimentos tardan entre 33,4 y 49,3 segundos. La media de la distribución es de 41,3 segundos. Además, en la Figura 9, se puede observar de una manera gráfica la distribución del tiempo que han tardado en ejecutarse todos los experimentos.

TABLA 2
VALORES RESUMEN DEL TIEMPO QUE TARDA EN CONVERGER EL ALGORITMO.

Medida	Tiempo (s)
Media	41,3
Desviación Estándar	13,1
Primer Cuartil (Q1)	33,4
Mediana	39,2
Tercer Cuartil (Q3)	49,3
Mínimo	11,0
Máximo	135,0

En la distribución de la Figura 9 se puede apreciar que los valores están muy concentrados en torno a la media. La distribución es leptocúrtica, es decir que hay mucha concentración. Además, presenta asimetría positiva ya que la mediana es menor que la media. Estos resultados respecto al tiempo hacen poco probable que el algoritmo pueda ser utilizado en dispositivos móviles en tiempo real. En el tiempo de ejecución del algoritmo influye el número de parámetros a optimizar y el rango de estos parámetros.

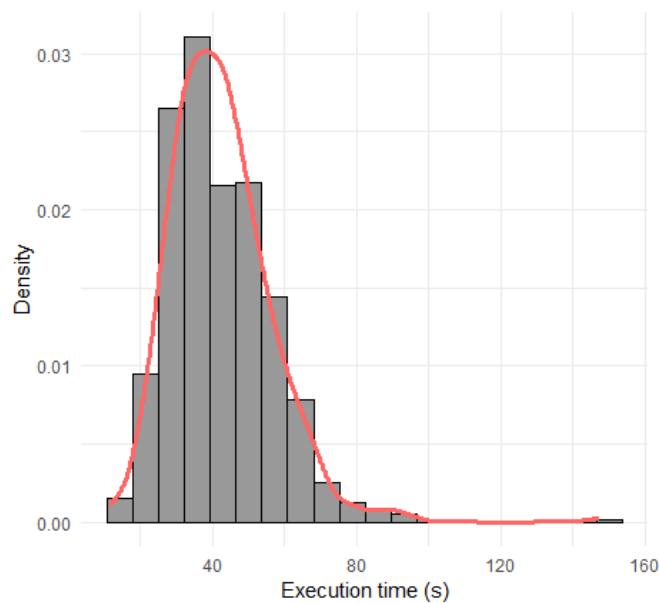


Figura 9. Distribución del tiempo de ejecución del algoritmo.

También se ha estudiado las puntuaciones obtenidas antes y después de aplicar la optimización. Para realizar esta comparación se han guardado las puntuaciones obtenidas por el algoritmo antes de aplicar ningún cambio sobre la foto y las puntuaciones obtenidas después de aplicar todos los cambios. El gráfico de dispersión de la Figura 10 compara la mejora obtenida en la puntuación de las imágenes en relación con su puntuación original. Se puede observar que hay una relación decreciente entre estas variables y que conforme la puntuación original aumenta la mejora obtenida por el algoritmo disminuye. Esto puede deberse a las limitaciones de la red neuronal y a que las puntuaciones obtenidas suelen estar en torno a la media de los datos. Como se ha explicado anteriormente, esta limitación se debe a que los datos utilizados en el entrenamiento no están balanceados y como consecuencia la red neuronal no predice puntuaciones por encima de siete y medio.

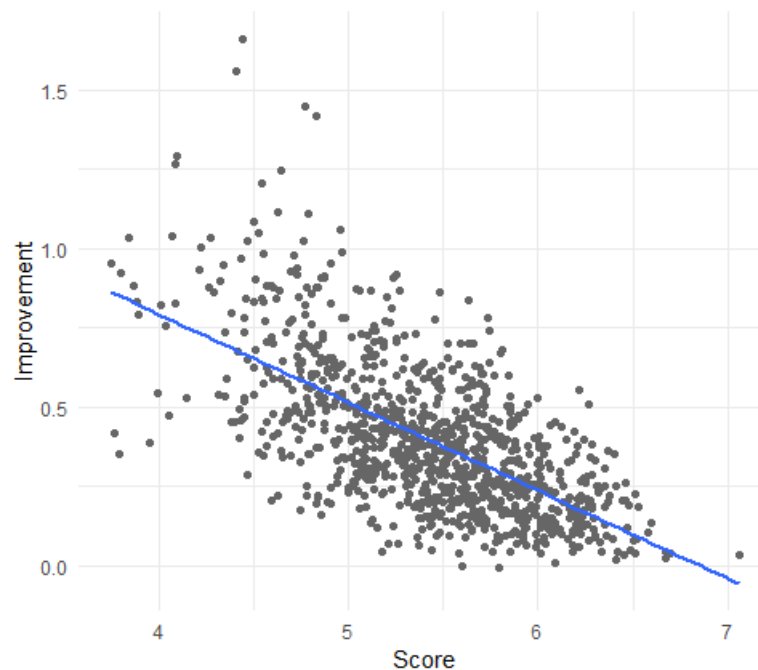


Figura 10. Relación entre mejora de la puntuación de las imágenes y puntuación original

La mayoría de las imágenes han mejorado después de aplicar el algoritmo de optimización. La Tabla 3 expone los estadísticos resumen del incremento de puntuación conseguido tras la aplicación del algoritmo. La mejora media en la puntuación de las imágenes ha sido de 0,398 puntos con una desviación estándar de 0,237 puntos. La mejora máxima ha sido de 1,66 puntos. La imagen que ha sufrido dicha mejora tenía una puntuación de 4,44 puntos antes de la utilización del algoritmo y una puntuación final de 6,10 puntos después. La mejora mínima ha sido, de hecho, un decremento de la

puntuación de la imagen de 0,006. Este hecho solo ha ocurrido en dos ocasiones y las imágenes han empeorado su puntuación de manera insignificante.

TABLA 3
MEDIDAS DESCRIPTIVAS DE LA MEJORA DE LAS IMÁGENES

Medida	Valor
Media	0,398
Desviación Estándar	0,237
Mínimo	-0,006
Máximo	1,661

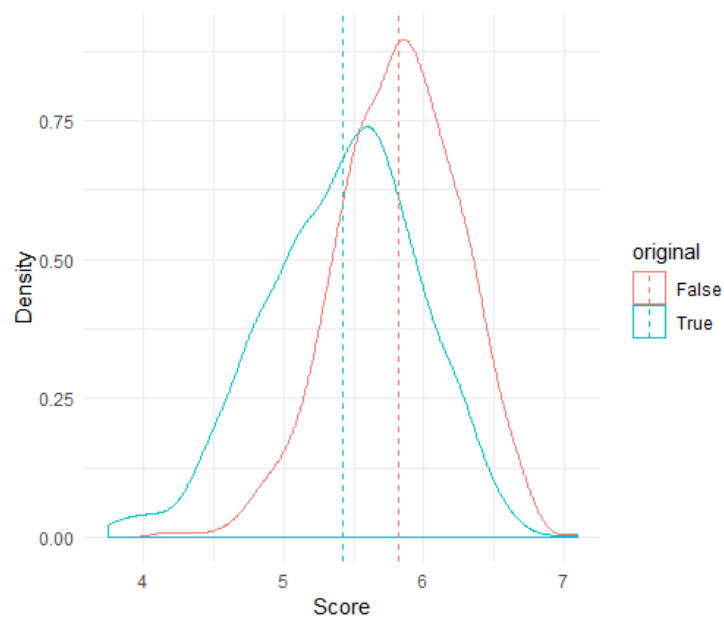


Figura 11. Distribución de las puntuaciones antes (azul) y después (rojo) de ejecutar el algoritmo de optimización.

La Figura 11 muestra la distribución de puntuaciones antes y después de que se haya utilizado el algoritmo. La media de puntuaciones de las imágenes originales es de 5,42, mientras que después de aplicarse el algoritmo la puntuación media asciende a 5,82. En la gráfica se puede apreciar como la distribución de densidad de las puntuaciones de las fotografías mejoradas está más concentrada en torno a la media y que es prácticamente

simétrica. Por el contrario, la distribución de densidad de las puntuaciones originales tiene mayor variabilidad y presenta una ligera asimetría negativa.

Por último, se ha observado cuales han sido los parámetros que más han sido modificados por el proceso de optimización. En la Figura 12 se muestran con gráficos de cajas y bigotes la distribución de los cambios realizados por el algoritmo sobre los distintos parámetros. Se puede observar que, aunque existe una tendencia a modificar más o menos algunos de ellos, la distribución presenta bastante variabilidad. Esto quiere decir que los filtros aplicados dependen en gran medida de la imagen original y su puntuación, por lo que el problema no se podría resolver aplicando el mismo filtro para todas las imágenes.

Los parámetros que más variabilidad tienen son el color y la nitidez. Esto quiere decir que el algoritmo encuentra valores muy distintos con alta frecuencia para aplicar estos filtros. En el caso opuesto se encuentra el brillo, que está muy concentrado. La velocidad del algoritmo se podría mejorar si se reduce el espacio de búsqueda del brillo solamente a aquellos valores más frecuentes.

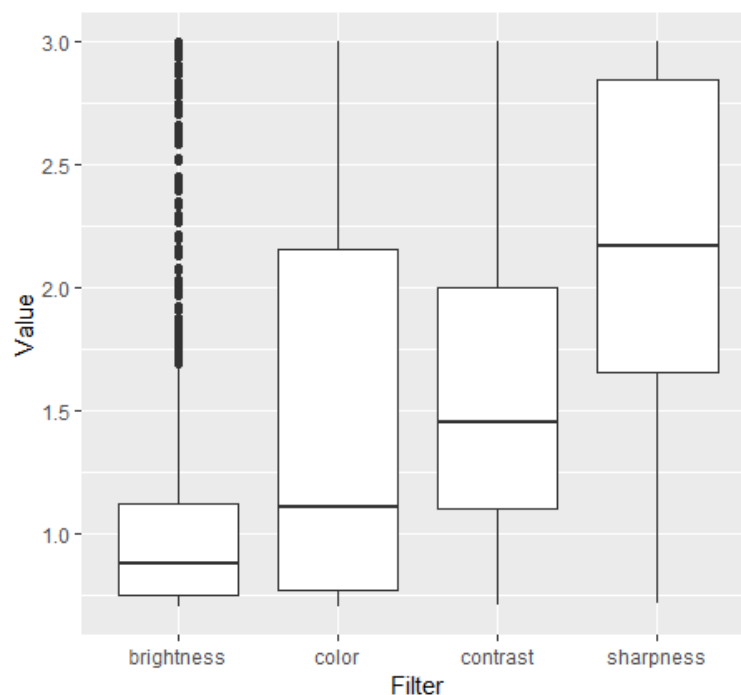


Figura 12. Diagrama de cajas que representa la distribución de los filtros aplicados para todas las imágenes.

A continuación, se muestran algunos ejemplos de imágenes antes y después de ser procesadas por el algoritmo del presente trabajo.

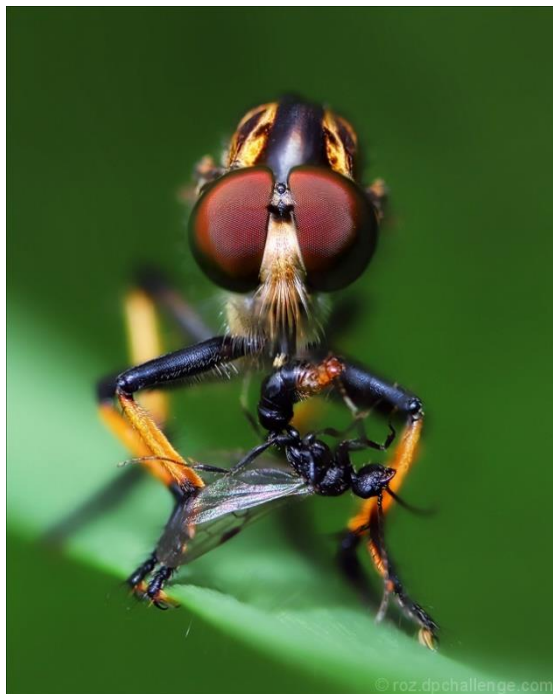


(a) Antes de la mejora. Puntuación (5,67)



(b) Después de la mejora. Puntuación (6,01)

Figura 13. Comparación de dos imágenes antes y después de aplicar el algoritmo. Los valores de modificación son los siguientes. Contraste (1,16), Brillo (1,14), Nitidez (1,59), Color (0,97)



(a) Antes de la mejora. Puntuación (6,20)



(b) Después de la mejora. Puntuación (6,36)

Figura 14. Comparación de dos imágenes antes y después de aplicar el algoritmo.

Los valores de modificación son los siguientes. Contraste (0,98), Brillo (1,49), Nitidez (1,97), Color (1,16)



(a) Antes de la mejora. Puntuación (3,88)



(b) Después de la mejora. Puntuación (4,71)

Figura 15. Comparación de dos imágenes antes y después de aplicar el algoritmo.

Los valores de modificación son los siguientes. Contraste (2,99), Brillo (0,80), Nitidez (2,85), Color (0,74)



(a) Antes de la mejora. Puntuación (7,05)



(b) Después de la mejora. Puntuación (7,09)

Figura 16. Comparación de dos imágenes antes y después de aplicar el algoritmo.

Los valores de modificación son los siguientes. Contraste (1,06), Brillo (0,86), Nitidez (1,46), Color (2,23)

4. CONCLUSIONES

As it has been discussed in the first part of this work, there are many studies and algorithms focused on improving the aesthetic quality of digital images. In recent times, thanks to the advancement of machine learning and convolutional neural networks, new methodologies have been found to carry out the process of improving digital images. This project uses the combination of a neural network that scores the aesthetics of the images and a differential evolutionary algorithm. The neural network is the "fitness" function of the evolutionary algorithm and the parameters to be optimized are a set of four photographic filters that control the color, brightness, contrast and sharpness of a photograph.

The main objective has been fulfilled. An algorithm that can make aesthetic improvements of digital images by using variable filters has been implemented. The variability of the filters is an important aspect because, otherwise, the resolution of the problem would be trivial. The images show improvement 99.8% of the time, although as shown in Figura 10 this improvement is greater when the image score is lower. The algorithm achieves an average improvement of 0.39 points per image. The intensity of the filters presents variability as seen in Figura 12, although the intensity of application of these varies with the type of filter used.

Using the algorithm in a mobile device to improve the images taken by the phone in real time is not viable currently. The average time it takes to improve a photo is 41.3 seconds. This time is too high to meet users' requirements. The average time of optimization could be reduced by using specific hardware architectures, such as TPUs, or by reducing the search space of the filters. For the first option, it would be necessary to develop an implementation of the differential evolution algorithm using a framework that allows tensor operations such as Tensorflow.

The AVA dataset may become a limitation when making improvements to high-quality images. The distribution of the scores of the data set is very unbalanced and centered, most of the scores are between 4 and 6. A dataset that includes images with better quality and has a more uniform distribution of scores could help to obtain better results. The algorithm that predict the score of the images could offer a greater range of scores and will be able to distinguish between a very low-quality picture and a very high quality one. It has not been possible to find a dataset with these characteristics and the creation of one was beyond the scope of this work.

Moreover, it should be noted that the flexibility of the proposed solution opens a many different options to improve in which further research could be carried out to improve the algorithm's result. This possible work that can be done in the future is explained below.

Firstly, and as has been mentioned previously, a tensorial implementation of the differential evolutionary algorithm could be implemented in order to speed up the processing of the images. Having low inference time is critical to use the algorithm in smartphones.

The second proposal is proposed to make an assessment of the images taking into account the category of these. This would allow more precise prediction. The colors brightness and contrast of the images vary according to their category and therefore it will be possible to apply more accurate improvements. Also knowing the semantic content of the images could be of great help.

Thirdly, different filters could be used to those proposed in this work. The algorithm is not dependent to any specific filter so trying out might be better. The use of filters like those used in photo editing programs such as Adobe Lightroom could generate better results, although it could slow down the optimization process.

Finally, as suggested before, improving the dataset used to train the neural network could have a huge impact in the score predictions made by the model. This would result in a more accurate assessment of the images and hence, in a better aesthetic improvement.

The proposed method has been proven to work correctly obtaining good results and meeting the main objective.

A shorter version of this project done in collaboration with Miguel Ángel Patricio, Antonio Berlanga and Jose Manuel Molina has been accepted in The International Conference on Hybrid Artificial Intelligent Systems 2019 organized by University of León.

5. PLANNING AND BUDGETING

A company willing to implement a solution based on this study, should consider many different factors. In this section it will be analyzed a hypothetical project that build a project on top of the algorithm proposed in this work. As mentioned earlier in the conclusion, further research would be necessary to be able to implement this algorithm on a mobile device that processes the images in real time.

The company might consider hiring talented people in order to develop the project. Two different profiles should be considered.

- 2 data scientists: Their responsibility will be continuing the project, improvement the functionality. They will consider doing the proposed future work as it could be helpful. They will need to have machine learning, python and devops knowledge. Their most important task will be reducing the inference time of the algorithm.
- 2 software engineers: Their responsibility will be modifying the smartphone's native camera libraries, so they will use the new algorithm. The most experienced engineer will become project manager and should take care of deadline and overview the project development.

The agile scrum framework will be used to develop the project. The task presents a lot of uncertainty and scrum provides a better way to handle changing scenarios than other traditional software engineering frameworks such as Waterfall. Each sprint will be of two weeks length and the product will be iterating until it meets all the requirements made by the team or the client. This methodology is the most appropriated for this kind of project because it allows to handle uncertainty.

The project will be composed of two stages. The first stage will be an initial planning, and after that there will be as many two weeks sprints as needed. The length of the project will be determined during the planification stage. Each sprint is made of four different task which are analysis, building, delivering and retrospecting.

Figure 17 shows a Gantt diagram with the planification stage and two iterations for a total of six weeks work. This diagram shows the main tasks, their dependencies and the length of them. Scrum is an iterative process, so each sprint has the same main tasks. Moreover, it has been decided that all employees work together, sharing insights to get a better understanding of the problem and nurturing collaborative spirit.

It is also important to analyze the project costs. Those can be divided in two main groups. In one hand there are employees' salaries and in the other hand there are material costs. Glassdoor [40] data has been used to make an estimation of the cost of each employee per year. Gross annual salary for a data scientist in Madrid is between 30,000.00 and 40,000.00 euros and for a software engineer is between 25,000.00 and 35,000.00 euros. Table 4 shows employee expenses using the average salary of the proposed interval. The total amount is the cost that the company has because gross annual salary has been used.

TABLE 4
EMPLOYEES' GROSS ANNUAL SALARY

Employee	Salary	Number	Total
Data Scientist	35.000,00 €	2	70.000,00 €
Software Engineer	30.000,00 €	2	60.000,00 €
Total			130.000,00 €

The total amount spent on salaries will depend on the number of months the project will take to finish. If the company already has employees, the amount may vary and should be used carefully.

The company will have other costs apart from the salaries of the employees. They will have other kind of costs such as electronic devices, working spaces and program licenses. This kind of costs can be split in fixed assets and expenses. The fixed assets will have a five years depreciation according to Spanish law. Table 5 shows the fixed assets that the company may incur to start with the project and its depreciation rate.

TABLE 5
FIXED ASSETS

Description	Unit price	Depreciation/yr	Units	Total
Computer Lenovo T580	1,150.00 €	230.00 €	4	920.00 €
Smartphone BQ X Pro	280.00 €	56.00 €	4	224.00 €
Total				1,144.00 €

Most of the expenses the company will have will be related to the training process of the machine learning models, because they are very computationally intensive. It is advisable to use cloud services like Google Cloud Platform to avoid buying expensive hardware and pay only for the resources that are being used. Moreover, if the company does not have its own offices, they could rent working spaces at WeWork [41] that would allow them to grow with flexibility.

TABLE 6
ESTIMATED ANNUAL EXPENSES

Description	Monthly	Units	Total
Google Cloud Platform	150.00 €	1	1,800.00 €
WeWork working space*	400.00 €	4	19,200.00 €
		Total	21,000.00 €

* In the case of not having already working spaces

Putting together all the expenses, depreciation and salaries, the annual price of the project would be 152,144.00 €, so the monthly cost would be around 12,700.00 €.

Due to the uncertainty of the project it is difficult to predict accurately the total length of it. During the planification stage the team should estimate the duration of the project, although this figure might change.

REFERENCIAS

- [1] Ibis World, "Camera Equipment Manufacturing in China," *IbisWorld*, May.
- [2] IMARC Group, "Digital Camera Market: Global Industry Trends, Share, Size, Growth, Opportunity and Forecast 2018-2023," February.
- [3] S. Sabin, "Smartphone Owners Prefer Simple Features Like Battery Life, Durability, Camera Quality,".
- [4] Parlamento Europeo, Consejo de la Unión Europea, "Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE," April.
- [5] Gobierno de España, "Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales," December.
- [6] Comisión Europea, "Comunicación de la Comisión al Parlamento Europeo, al Consejo Europeo, al Consejo, al Comité Económico y Social Europeo y al Comité de las Regiones. Inteligencia artificial para Europa," April.
- [7] H. Han and K. Sohn, "Automatic illumination and color compensation using mean shift and sigma filter," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 978-986.
- [8] A.K. Jain, "Fundamentals of digital image processing," 1989.
- [9] J.A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Trans.Image Process.*, vol. 9, no. 5, pp. 889-896.
- [10] S.K. Naik and C.A. Murthy, "Hue-preserving color image enhancement without gamut problem," *IEEE Trans.Image Process.*, vol. 12, no. 12, pp. 1591-1598.
- [11] G. Song and X. Qiao, "Color image enhancement based on luminance and saturation components," vol. 3, pp. 307-310.
- [12] A. Krizhevsky, I. Sutskever and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," pp. 1097-1105.
- [13] D. Strigl, K. Kofler and S. Podlipnig, "Performance and scalability of GPU-based convolutional neural networks," pp. 317-324.
- [14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," pp. 1-9.
- [15] L.A. Gatys, A.S. Ecker and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*.

- [16] A. Huang and R. Wu, "Deep learning for music," *arXiv preprint arXiv:1606.04930*.
- [17] Z. Wang, S. Chang, F. Dolcos, D. Beck, D. Liu and T.S. Huang, "Brain-inspired deep networks for image aesthetics assessment," *arXiv preprint arXiv:1601.04155*.
- [18] X. Lu, Z. Lin, H. Jin, J. Yang and J.Z. Wang, "Rating image aesthetics using deep learning," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2021-2034.
- [19] Y. Kao, C. Wang and K. Huang, "Visual aesthetic quality assessment with a regression model," pp. 1583-1587.
- [20] H. Talebi and P. Milanfar, "NIMA: Neural image assessment," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3998-4011.
- [21] Y. Deng, C.C. Loy and X. Tang, "Aesthetic-driven image enhancement by adversarial learning," pp. 870-878.
- [22] H. Fang and M. Zhang, "Creatism: A deep-learning photographer capable of creating professional work," *arXiv preprint arXiv:1707.03491*.
- [23] X. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE access*, vol. 2, pp. 514-525.
- [24] K. Lasse Lueth, "State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating,".
- [25] N.P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden and A. Borchers, "In-datacenter performance analysis of a tensor processing unit," pp. 1-12.
- [26] N. Murray, L. Marchesotti and F. Perronnin, "AVA: A large-scale database for aesthetic visual analysis," pp. 2408-2415.
- [27] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J. Astola, B. Vozel, K. Chehdi, M. Carli and F. Battisti, "Color image database TID2013: Peculiarities and preliminary results," pp. 106-111.
- [28] S. Kong, X. Shen, Z. Lin, R. Mech and C. Fowlkes, "Photo aesthetics ranking network with attributes and content adaptation," pp. 662-679.
- [29] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optimiz.*, vol. 11, no. 4, pp. 341-359.
- [30] P. Rodríguez-Mier, "A tutorial on Differential Evolution with Python,".
- [31] K. Price and R. Storn, "Differential Evolution (DE) for Continuous Function Optimization,".

- [32] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*.
- [33] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," pp. 770-778.
- [34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*.
- [35] A. Dertat, "Applied Deep Learning - Part 4: Convolutional Neural Networks,".
- [36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*.
- [37] Y. LeCun, D. Touresky, G. Hinton and T. Sejnowski, "A theoretical framework for back-propagation," vol. 1, pp. 21-28.
- [38] Y. Rubner, C. Tomasi and L.J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, no. 2, pp. 99-121.
- [39] C. Lennan, H. Nguyen and D. Tran, "Image Quality Assessment,".
- [40] Anonymous "Glassdoor,". Available: <https://www.glassdoor.com>.
- [41] Anonymous "WeWork". Available: <https://www.wework.com/>.